Conservative Unsteady Simulation of Arbitrary Motion in Two-dimensional Spacetime Using Central Difference, Upwind and Hybrid Formulations

I. Flamarique Ederra * ; T. C. S. Rendall † ; A. L. Gaitonde ‡ ; D. Jones § ; C. B. Allen ¶

Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, UK

A spacetime framework is presented to solve unsteady aerodynamics problems as an alternative to conventional approaches for complex problems involving large deformation or topological change such as store separation, slat and flap deployment or spoiler deflection. It avoids complex CFD meshing methods, such as Chimera, by the use of a finite-volume approach both in space and time. The use of a central-difference scheme in the time direction yields non-physical transient solutions as a consequence of pressure waves travelling backwards in time. Therefore, an upwind formulation is provided and validated against one-dimensional test cases: a semi-infinite piston and a finite piston with a sharp change in direction. Also a hybrid formulation is given and several two-dimensional unsteady aerodynamics cases are computed with all three formulations and compared, demonstrating that the use of an upwind time stencil yields more representative physical solutions and improves the rate of convergence. In particular, the following problems are presented: a pitching NACA-0012 (periodic); a simple flap deflection; a spoiler deployment; a full landing case with a combination of slat, flap and spoiler deployments along with ground effect; and a case where aerofoils fly in opposite directions at subsonic and supersonic speeds.

Nomenclature

a	=	Speed of sound
α	=	Angle of attack
b	=	Aerofoil's semi-chord
E	=	Total energy of fluid
$\mathbf{e}_i$	=	Unit vector along i direction
γ	=	Heat capacity ratio of fluid
F	=	Matrix of fluxes
$\mathbf{f}_i$	=	Column vector of fluxes in i
k	=	Reduced frequency
φ_i	=	Slope limiter along i direction
M_n	=	Normal Mach number
$\mathbf{n}$	=	Normal vector
n_i	=	Normal vector component in i
n_{ne}	=	Number of neighbouring cells
∇	=	Divergence operator
p	=	Pressure of fluid
$\mathbf{R}$	=	Column vector of residuals
r_j	=	Monitor of slope limiter at cell j
ρ	=	Density of fluid
S	=	Surface area
S_i	=	Projection of surface area normal to i
t	=	Time coordinate
t^*	=	Pseudo-time coordinate
θ	=	Angle of deflection
ω	=	Angular velocity, angular frequency
JST	=	Jameson-Schmidt-Turkel (central-difference)
CSUT	=	Central in Space, Upwind in Time
$\mathbf{U}$	=	Column vector of conserved variables
u_n	=	Normal velocity
u, v, w	=	Fluid velocities along x, y and z
V	=	Control volume
$\mathbf{W}$	=	Column vector of primitive variables
x, y, z	=	Spatial coordinates

Subscripts and superscripts

$+$	=	Upstream
$-$	=	Downstream
c	=	Cell centre
f	=	Face centre
n	=	Normal or perpendicular direction
ST	=	Spacetime
t	=	Time direction
τ, t_1, t_2	=	Tangential directions
VL	=	Van Leer
x, y, z	=	Spatial directions

*PhD Student, AIAA Student Member, imanol.flamarique@bristol.ac.uk, University of Bristol, Bristol, BS8 1TR, UK

†Lecturer, AIAA Member, thomas.rendall@bristol.ac.uk, University of Bristol, Bristol, BS8 1TR, UK

‡Senior lecturer, AIAA Senior Member, ann.gaitonde@bristol.ac.uk, University of Bristol, Bristol, BS8 1TR, UK

§Senior lecturer, AIAA Senior Member, dorian.jones@bristol.ac.uk, University of Bristol, Bristol, BS8 1TR, UK

¶Professor of Computational Aerodynamics, AIAA Senior Member, c.b.allen@bristol.ac.uk, University of Bristol, Bristol, BS8 1TR, UK

I. Introduction

The study of interaction between helicopter rotor-blades and a fuselage constitutes a clear example of the complexity of unsteady aerodynamics. Other common complex problems include store separation, flap and spoiler deployment in take-off and/or landing configurations or the transient process that takes place within an internal combustion engine when valves open/close. Finding accurate and efficient solutions to these problems using the most common CFD solvers remains limited by the capability of existing mesh generation/deformation techniques and interpolation algorithms. A different meshing technology needs to be used depending on the problem under consideration which, inevitably, limits the ability to automate simulations and slows down the design cycle of industrial applications.

Although an integration across the four-dimensional space-time domain is required to obtain unsteady solutions of the three-dimensional Euler equations, conventional methods have traditionally decoupled this process into two consecutive and different steps: a finite-volume integration in a three-dimensional space and a finite-difference integration in time. The novelty introduced by the spacetime method implemented here is that both integrations, in space and time, are treated similarly through the use of spacetime finite-volumes, hence effectively solving unsteady problems of dimension N as steady problems of dimension $N + 1$. This implies modifying the fluid equations of motion through the divergence theorem to remove temporal derivatives, which are replaced by temporal fluxes instead.

However, coupling time and space, and effectively solving both as one, may lead to non-physical behaviour when using a central-difference scheme since information can be propagated backwards in time as a consequence of the temporal stencil being used [1]. The solution at a certain time level may be affected by the solution at later times. In the present paper, an upwind stencil is used instead to solve this issue. Several solutions demonstrate that the use of an upwind scheme yields more representative solutions compared to central-difference solutions. Some solutions from a hybrid central-difference/upwind scheme (CSUT) are also given.

II. Background

A. Existing unsteady CFD meshing methods

Historically, the simulation of unsteady aerodynamic problems has been restricted to existing techniques such as mesh motion, Chimera grids or immersed boundary methods amongst others, that allow volume meshes to accommodate surface mesh motions. These methodologies must be used along with an Arbitrary Lagrangian-Eulerian (ALE) formulation of the fluids equations [2] and come with limitations in regard to the type of motions they can cope with. In general, mesh deformation techniques can only deal with small movements if a good quality mesh is to be retained after the deformation process. The fact that no cells can appear or disappear due to a fixed connectivity between cells at two consecutive time levels yields distorted cells with high aspect ratios when large body motions are involved. In such cases, re-meshing, i.e. generating a completely new mesh from the geometry at the current time level would be a suitable solution to the problem of low quality meshes. However, this implies using an interpolation method to relate the fluid variables in the new mesh with those in the previous one, hence making a very intensive usage of available computational resources. Interpolating is not a trivial task, can be non-conservative and often demands a degree of ingenuity.

An automatic mechanism is sought in order to create good quality meshes as an adaptation from previous grids, without the user intervention. One of the simplest and most widely used methods is mesh motion whereby the grid nodes move as a consequence of the moving domain boundaries. Their simplicity leads to a low computational cost and the fact that the nodal connectivity list remains unaltered through time marching, i.e. there is a one-to-one mapping between cells of two successive time levels, hence enabling the use of the last computed solution directly without the need for interpolation. It is interesting to look at the analysis and formulation of robust mesh motion techniques implemented by Mavriplis et al. [3–5]. The spring analogy [6–9] compares the mesh to a system of tensional and/or torsional springs whereby a displacement of the boundary forces the movement of interior nodes in order for the system to stay in equilibrium. Although the exclusive use of tensional springs is usually enough, the addition of torsional springs is advisable to minimize the chances of getting distorted meshes, but this comes at the expense of a greater computational cost. Transfinite interpolation is based on the definition of a bilinearly blended interpolant which maps the domain boundaries at a non denumerable number of points [10, 11], i.e. a curve in a two-dimensional space or a surface in a three-dimensional one. It is very efficient, requiring small computational power, and works well with structured grids provided that boundaries do not get heavily distorted [10,11]. Radial basis functions (RBF) [12,13] are extremely versatile and cope well with fluid-structure interaction and relatively large mesh motion problems [14, 15]. Their value depends only upon the distance to reference points and is used to define an interpolation function in a similar fashion as Taylor series use function's derivatives at a point to approximate the function in the nearby. Mesh motion techniques that exploit RBF's are efficient and inexpensive since the coupling matrix is calculated only once, with no further modifications needed thereafter, and all other operations being simple matrix multiplications. An

optimal selection of the reference points or centres can be obtained effectively through a greedy method [16–18] which improves greatly the efficiency by finding redundant centres. Finally, Laplacian smoothing attempts to fix the mesh and recover the original grid quality via finite differences of Laplace's equation [19] when large displacements of boundary nodes affect the quality of cells nearby. An improved and more expensive version of the method called smart Laplacian smoothing [20] can prevent cell inversion by checking whether the new node locations degrade mesh quality.

When complex rotational parts or relative motions [21] are involved, Chimera or overset grids become a more reasonable alternative to mesh motion techniques thanks to the use of a separate body-fitted mesh for each of the moving parts. A global volume mesh is also defined in the background and an intersection between them is performed to find the interfaces at each time level. This allows splitting of the fluid domain into several regions and discretisation of them independently, effectively transforming a very intricate mesh generation problem into several simpler and smaller ones. In addition, simple yet powerful high-quality structured grids can be used around each of the moving parts, which translates into more efficient and faster fluid solvers and mesh generators [22, 23]. Boundary motions are very much simplified and only a rotation and/or translation of the existing grids is required before the intersection process happens again, hence saving computational effort. Due to their block-based nature, Chimera grids are suitable for parallel computations [24] and represent a reasonable alternative when an adaptive refinement mechanism is to be implemented [22]. Despite all of the above, interpolation algorithms needed at the boundaries of two overlapping grids are usually costly and complex [23], and can introduce numerical errors unless special care is taken to minimize them. They still cannot deal with arbitrary motions such as aeroelastic problems [24], where a mesh deformation technique is required in addition, or situations involving topological changes with appearing/disappearing cells, which, once again, rely on interpolations of the solution.

As opposed to Chimera, sliding grid planes are based on grids whose boundaries fit together without any overlapping at all, and slide past each other when there exists a relative motion. An interpolation method must still be put into place in order to communicate flow variables at both sides of the interface [25, 26]. A method for the study of helicopter rotor-fuselage interaction is proposed by Steijl et al. [26] proving its accuracy and efficiency, provided the mesh size is not too big, but performing poorly under parallel computations. Moreover, limitations regarding the allowable timestep are also important. Rumsey [25] showed that a small timestep is crucial for the construction of an interpolation scheme that deals properly with the propagation of acoustic waves across sliding planes. Likewise, the work of Fenwick et al. [27] led to the conclusion that, during the maximum timestep, grids should slide no more than the size of one cell if a correct unsteady behaviour is to be captured.

Immersed boundary methods [28–30] or Cartesian cut-cell grids [30, 31] can also be a feasible alternative to deal with mesh deformation in unsteady aerodynamics. Both rely on a fixed Cartesian grid in the background. In the former, the fluid solver is fully responsible for modelling the boundaries [28, 29], leaving the mesh unaltered and leading to inaccurate surface mesh representation for compressible flows, whereas in the latter, the actual grid is cut along the boundaries at every time level [30], incurring in a relatively high computational cost. Immersed boundary methods often fail to ensure conservation of mass, momentum or energy in cells cut by a solid boundary and it is therefore not a popular technique across the aerospace industry where compressible aerodynamics demand a good quality representation of boundaries. Moreover, very thin boundary layers cannot be captured sensibly unless either the cell count is high, meaning large and expensive simulations, or anisotropic refinement is used [32]. Cut-cell grids also require interpolations between consecutive time levels.

Meshless methods, in an attempt to circumvent what appears to be the main bottleneck for industrial application of numerical techniques, i.e. good quality and automated mesh generation for complex geometries with sharp edges [33], replace the traditionally used grid by a dense cloud of points based on which conservation laws can be discretized [34]. Whilst connectivity information is inherently lost, more programming effort is needed compared to traditional mesh-based methods since there is still the need for finding the neighbours which lie within the domain of influence of each node [35], which can be a very time consuming task. Moreover, the generation of optimal distributions of points tailored for each specific problem is not a trivial task [33, 36] and conservation can sometimes be an issue. Lagrangian-smoothed particle hydrodynamics (SPH) [33, 35], meshfree methods such as finite points [37] or meshless local Petrov-Galerkin method are good examples of this.

As outlined before, re-meshing can deal successfully with any arbitrary motion preserving good quality meshes at all times [38–40]. Compared to the above methods it is likely to be the most demanding one in terms of computational effort. Being capable of dealing with structured grids, it is with unstructured meshes where the greater gains in efficiency are achieved given the ability to modify only certain regions of the domain. Nevertheless, it is always necessary to work out the connectivity relationship between cells at consecutive time levels and an appropriate interpolation method has to be derived in the event of topological changes such as appearing/disappearing cells, which may introduce numerical errors across the solution.

The spacetime framework offers an alternative conservative simulation approach even with topological changes and variable real time-steps and if appropriately implemented can preserve time accuracy.

B. Spacetime method

An early spacetime formulation came from Giles [41] in 1988. The imposition of periodic boundary conditions in turbomachinery flows is a complex task, especially when the rotor and stator have different pitch values (i.e. distance between blades). By inclining the computational time plane Giles circumvents this issue and transforms the Euler equations so that any stator-rotor pair can be treated with a pitch ratio of 1. At the same time, Hughes et al. [42] apply a spacetime technique to classical elastodynamics problems via the use of a finite element approach with a discontinuous Galerkin (DG) formulation in the time direction. Lowrie et al. [43] build on the previous work for a much more general problem involving hyperbolic conservation laws. Again, they use a discontinuous Galerkin formulation to create a higher order scheme within the spacetime framework. However this results in a computationally expensive method compared to other conventional approaches. Moreover, their work implicitly assumes some regularity on the structure of the grids used, hence invalidating a more general boundary motion approach. Thompson et al. [44] and Ray [45] use a DG formulation to give their own interpretation of the spacetime method. In particular, Thompson et al. [44] present an adaptive spacetime technique that allows refinement and coarsening of the grid and which they define as robust. This robustness comes at a sacrifice of the general applicability of the method since they retain orthogonal planes in the time direction leaving the time integration fully decoupled from the space integration.

Tsuei et al. [46] successfully apply the spacetime method developed earlier by Chang [47–49] at NASA to blade row interaction problems in turbomachinery flows. The so-called space-time conservation element and solution element method (CE/SE) is able to predict unsteady flows without any previous assumptions imposed on the solver, by simply considering fluxes both in space and time. They argue that the scope of this new method is large and that a wide range of applications can benefit from it, however their work is focused on turbomachinery applications and no attempts made towards a more general and arbitrary motion. Perhaps the most general implementation of the spacetime method are the works by Hixon [50, 51] and Golubev et al. [52, 53]. Their method allows for a variable timestep size across the fluid domain and no decoupling is made between temporal and spatial integrations, allowing for higher-order schemes to be used in the time integration. Zwart et al. [54] apply the spacetime formulation to the solution of a breaking dam, although their implementation lacks a general spacetime mesh with varying timestep sizes across the spatial domain. Similarly, Van der Ven [55] applies a conservative adaptive multigrid algorithm under the spacetime framework to investigate an oscillating two-dimensional aerofoil, demonstrating the potential of the method.

More recently, Rendall et al. [1, 56–58] use a general formulation of the spacetime method and show its capability for simulating complex moving geometries in one and two-dimensional spaces. In their work they compare their solution against analytical results available from piston theory and investigate the problem of a pitching NACA 0012 aerofoil with both structured and unstructured meshes. Due to the use of a central-difference scheme in time they observe a slight change in the behaviour of the pressure distribution with respect to a conventional dual time-stepping solver. They explain this phenomenon with information propagated backwards in time as a consequence of the temporal stencil being used. They also show the ability of the spacetime solver to cope with rotating parts and appearing/disappearing objects in both a stationary gas and moving subsonic and supersonic flows, as well as a problem like store separation.

III. Formulation

The numerical solution of the three-dimensional Euler equations for inviscid compressible flows requires integrating them both in space and time. Using the Green's or divergence theorem to convert a surface integral into a volume integral across the volume enclosed by the surface yields

$$\int_{t_0}^{t_F} \int_V \left(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} \right) d\Omega d\tau = \mathbf{0} \tag{1}$$

where the vector of conserved variables $\mathbf{U}$ is

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix} \tag{2}$$

and the matrix of fluxes $\mathbf{F}$ is

$$\mathbf{F} = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho uv & \rho uw \\ \rho uv & \rho v^2 + p & \rho vw \\ \rho uw & \rho vw & \rho w^2 + p \\ (\rho E + p)\, u & (\rho E + p)\, v & (\rho E + p)\, w \end{bmatrix} \tag{3}$$

The key aspect of the spacetime method is the treatment of the time integration identically to the space integration through the use of four-dimensional spacetime finite-volumes. Within this framework any unsteady problem of dimension N can be effectively solved as another steady problem of dimension $N + 1$. In the current three-dimensional problem, a four-dimensional divergence operator can be defined as (the subscript ST means spacetime)

$$\nabla_{\text{ST}} = \frac{\partial}{\partial t}\mathbf{e}_t + \frac{\partial}{\partial x}\mathbf{e}_x + \frac{\partial}{\partial y}\mathbf{e}_y + \frac{\partial}{\partial z}\mathbf{e}_z \tag{4}$$

which leads to the four-dimensional spacetime formulation of the Euler equations (1) as

$$\int_{V_{\text{ST}}} \nabla_{\text{ST}} \cdot \mathbf{F}_{\text{ST}}\, d\Omega_{\text{ST}} = \mathbf{0} \tag{5}$$

or expressed as a closed surface integral

$$\oint_{\partial V_{\text{ST}}} \mathbf{F}_{\text{ST}} \cdot \mathbf{n}_{\text{ST}}\, dS_{\text{ST}} = \mathbf{0} \tag{6}$$

where the matrix of spacetime fluxes $\mathbf{F}_{\text{ST}}$ is

$$\mathbf{F}_{\text{ST}} = \begin{bmatrix} \rho & \rho u & \rho v & \rho w \\ \rho u & \rho u^2 + p & \rho uv & \rho uw \\ \rho v & \rho uv & \rho v^2 + p & \rho vw \\ \rho w & \rho uw & \rho vw & \rho w^2 + p \\ \rho E & (\rho E + p)\, u & (\rho E + p)\, v & (\rho E + p)\, w \end{bmatrix} \tag{7}$$

and the spacetime normal vector $\mathbf{n}_{\text{ST}}$ is

$$\mathbf{n}_{\text{ST}} = \begin{Bmatrix} n_t \\ n_x \\ n_y \\ n_z \end{Bmatrix} \tag{8}$$

Equation (6) can be regarded as the integration of the steady Euler equations across a theoretical four-dimensional space and, as such, it can be solved by means of four-dimensional fluxes in a pseudo-time marching fashion, until residuals converge to zero

$$\frac{\partial V \mathbf{U}}{\partial t^*} + \mathbf{R}\,(\mathbf{U}) = \mathbf{0} \tag{9}$$

where the column vector of residuals is

$$\mathbf{R}\,(\mathbf{U}) = \oint_{\partial V_{\text{ST}}} \mathbf{F}_{\text{ST}} \cdot \mathbf{n}_{\text{ST}}\, dS_{\text{ST}} \tag{10}$$

Unlike commonly applied methods, the use of a finite-volume approach for the discretization in time, as well as in space, ensures the automatic conservation of mass, momentum and energy and, more importantly, allows the use of a variable real timestep across the spatial domain without causing a non-physical behaviour of the solution. Notice the potential gain in efficiency over conventional time-stepping techniques due to the fact that a bigger timestep can be used in areas of freestream flow, far away from the perturbations, whilst still retaining sufficiently small timesteps in areas where rapid changes occur. In terms of solution accuracy the spacetime method brings the possibility to incorporate some of the higher-order schemes used for spatial discretizations into the temporal dimension.

The spacetime framework works well with any arbitrary motion, from big boundary displacements, like a helicopter rotor blade, through to geometric topological changes such as a store separation or a slotted flap deflection. There is no need for further modifications to the solver in any of the former cases, mainly as a consequence of the finite-volume approach in time. All the information related to boundary motions is implicitly given by the spacetime

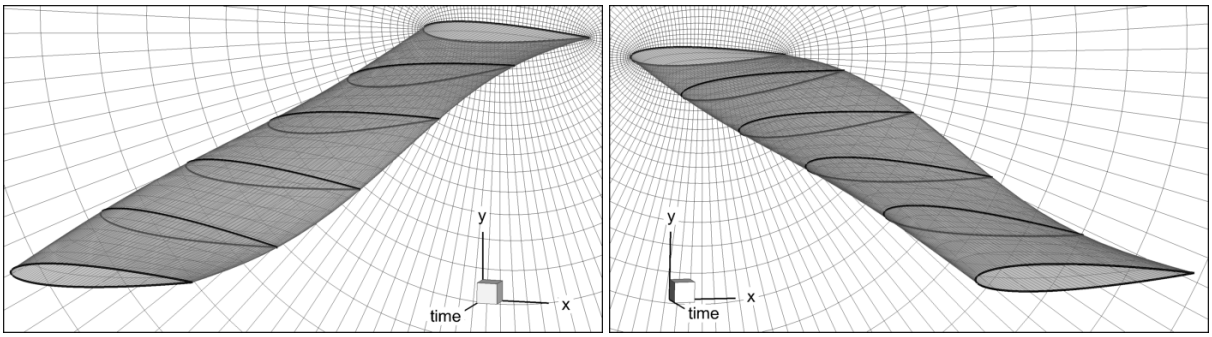


Figure 1. Example of spacetime mesh: pitching NACA 0012 aerofoil

mesh. An example of this is depicted in Figure 1 where the pitching movement of a NACA 0012 aerofoil is given by a twisted wing in which the span-wise direction represents the time. Moreover, no connectivity relationship is required between cells at different time levels[a] since this is implicitly accomplished by the spacetime mesh, therefore allowing for appearing/disappearing cells without the need for interpolation.

Along with the development of a spacetime framework there is a need for new grid generation techniques. Being able to generate unstructured grids in the time direction brings the possibility to refine the timestep size in some areas of the domain while keeping a coarse one in others where temporal resolution is not required. Currently, it is possible to use available 3D grid generators to create 2D unsteady meshes well-suited for the spacetime framework. However, there is no available technology to automatically generate a truly unstructured four-dimensional grid to be used in 3D+t problems. Although not yet mature, Behr [59] introduces a simple meshing technique that allows unstructured grids to be created, not only for 2D+t problems but also 3D+t. Likewise, Ungor et al. [60] and, some time later, Abedi et al. [61] have put some efforts towards the development of 2D+t grids by a mesh-marching technique.

IV. Flux-splitting

Using a dual time-stepping method [62], backward differences are implicitly being used for the time derivatives. This means that the solution at each time level depends only upon the solution calculated at previous time levels. Within the spacetime framework, however, there exists the possibility to use a central-difference scheme [1, 56–58] resulting in information being propagated backwards in time [1]. Namely, whatever happens in the future affects the solution in the past which, obviously, violates a principle of causality. Even with a central-difference scheme the results obtained for some test cases are remarkably close to the dual time stepping solutions.

A spacetime version of the upwind flux-splitting method proposed by Van Leer [63] is implemented in this work. The two-dimensional formulation is given below.

A. Two-dimensional upwind formulation

At any inclined face in a spacetime grid the total flux may be split into space and time contributions. In a two-dimensional case there will be three terms: two contained within the spatial plane (x, y) and one along the time direction t. Equation (10) can be cast to

$$\begin{aligned}\mathbf{R} = \sum_{\text{faces}} \mathbf{F}_{\text{ST}} \cdot \mathbf{n}_{\text{ST}} \cdot S = \sum_{\text{faces}} (\mathbf{f}_t \cdot n_t + \mathbf{f}_x \cdot n_x + \mathbf{f}_y \cdot n_y) \cdot S = \\ = \sum_{\text{faces}} \mathbf{f}_t \cdot S_t + \mathbf{f}_x \cdot S_x + \mathbf{f}_y \cdot S_y\end{aligned} \tag{11}$$

where $\mathbf{f}_t$ is the first column vector in the two-dimensional version of equation (7) which represents the time fluxes

$$\mathbf{f}_t = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix} \tag{12}$$

[a]Note that it would not be possible to talk about cells being at a certain time level since different cells span between different time levels.

and $\mathbf{f}_x$ and $\mathbf{f}_y$ are the second and third column vectors in the two-dimensional version of equation (7) which correspond to the space fluxes

$$\mathbf{f}_x = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho E + p)\, u \end{Bmatrix} \qquad \mathbf{f}_y = \begin{Bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (\rho E + p)\, v \end{Bmatrix} \tag{13}$$

As implied by a characteristic analysis, a pseudo-velocity that is constant and equal to one can be defined in the time direction, i.e. $u^* = \dfrac{dt}{dt^*} = 1$. This means that, for a physically meaningful solution, information in time is always convected from the upstream (or previous in time) cell which corresponds to that with the smallest t coordinate. Therefore time fluxes must be calculated using the primitive variables evaluated only at the upstream side of the face in equation (12), i.e.

$$\mathbf{f}_t\left(\mathbf{W}^+\right) = \begin{Bmatrix} \rho^+ \\ \rho^+ u^+ \\ \rho^+ v^+ \\ \rho^+ E^+ \end{Bmatrix} \tag{14}$$

where superscript $+$ denotes upstream and the column vector $\mathbf{W}^+$ of primitive variables is evaluated at the upstream side of the face

$$\mathbf{W}^+ = \begin{Bmatrix} \rho^+ \\ u^+ \\ v^+ \\ p^+ \end{Bmatrix} \tag{15}$$

The sign of $\mathbf{n} \cdot \mathbf{e}_t$ at each face may be used to discern between upstream and downstream cells in time (or past and future cells).

For fluxes in space information may be convected from both sides, upstream and downstream, if the flow is subsonic. This is consistent with the flux-vector splitting method developed by Van Leer [63] which allows to work out the contribution of each side of the face to the total flux.

Since only the component of the velocity normal to the face will give non-zero fluxes in the momentum equation, the local normal Mach number $M_n = u_n/\sqrt{\gamma p/\rho}$ is used. There are two possible cases. If the normal flow is supersonic ($|M_n| \geq 1$) fluxes are given by the properties at the upstream side only

$$\mathbf{f}_n\left(\mathbf{W}_n^+\right) = \begin{Bmatrix} \rho^+ u_n^+ \\ \rho^+ \left(u_n^+\right)^2 + p^+ \\ \rho^+ u_n^+ u_{t_1}^+ \\ \rho^+ u_n^+ u_{t_2}^+ \\ \left(\rho^+ E^+ + p^+\right) u_n^+ \end{Bmatrix} \tag{16}$$

whereas in the case of subsonic flow ($|M_n| < 1$) fluxes are formed by contributions from both sides, upstream ($+$) and downstream ($-$)

$$\mathbf{f}_n\left(\mathbf{W}_n\right) = \mathbf{f}_n^+\left(\mathbf{W}_n^+\right) + \mathbf{f}_n^-\left(\mathbf{W}_n^-\right) \tag{17}$$

where the normal flux functions $\mathbf{f}_n^+$ and $\mathbf{f}_n^-$ are given [63] in equation (18), as follows

$$\mathbf{f}_n^\pm\left(\mathbf{W}_n^\pm\right) = \pm\frac{\rho^\pm c^\pm}{4}\left(M_n^\pm \pm 1\right)^2 \begin{Bmatrix} 1 \\ \dfrac{c^\pm}{\gamma}\left[\left(\gamma - 1\right) M_n^\pm \pm 2\right] \\ u_{t_1}^\pm \\ u_{t_2}^\pm \\ \dfrac{\left(c^\pm\right)^2\left[\left(\gamma - 1\right) M_n^\pm \pm 2\right]^2}{2\left(\gamma^2 - 1\right)} + \dfrac{\left(u_{t_1}^\pm\right)^2 + \left(u_{t_2}^\pm\right)^2}{2} \end{Bmatrix} \tag{18}$$

and the column vectors of normal primitive variables, $\mathbf{W}_n^+$ and $\mathbf{W}_n^-$, at the upstream and downstream sides of the face, respectively, are

$$\mathbf{W}_n^+ = \begin{Bmatrix} \rho^+ \\ u_n^+ \\ u_{t_1}^+ \\ u_{t_2}^+ \\ p^+ \end{Bmatrix} \qquad \mathbf{W}_n^- = \begin{Bmatrix} \rho^- \\ u_n^- \\ u_{t_1}^- \\ u_{t_2}^- \\ p^- \end{Bmatrix} \tag{19}$$

Notice here the difference between functions $\mathbf{f}_n\left(\mathbf{W}_n^{\pm}\right)$ and $\mathbf{f}_n^{\pm}\left(\mathbf{W}_n^{\pm}\right)$. Also, bear in mind that u_n, u_{t_1} and u_{t_2}, given by (20), are the components of the real velocity $\mathbf{v} = \{0, u, v\}^T$ projected onto a spacetime coordinate system defined locally at each face such that axis n is normal to the face and the other two, t_1 and t_2, are tangential. Depending on the face orientation the new coordinates may not be purely spatial or purely temporal but a combination of spatial and temporal coordinates. In other words, the projection of the flow velocity $\mathbf{v}$, strictly defined in space (x, y), onto the local coordinate system yields components in spacetime, as follows,

$$\begin{Bmatrix} u_n \\ u_{t_1} \\ u_{t_2} \end{Bmatrix} = \begin{bmatrix} \\ P^T \\ \\ \end{bmatrix} \begin{Bmatrix} 0 \\ u \\ v \end{Bmatrix} = \begin{Bmatrix} un_x + vn_y \\ \dfrac{vn_x - un_y}{\sqrt{n_x^2 + n_y^2}} \\ \dfrac{n_t\left(un_x + vn_y\right)}{\sqrt{n_x^2 + n_y^2}} \end{Bmatrix} \tag{20}$$

where transformation matrix $P : (t, x, y) \mapsto (n, t_1, t_2)$, which maps the global space+time coordinate system to the spacetime coordinate system locally defined at each face, is given by equation (24). Note here the intended distinction between a space+time ($\mathbb{R}^2 \cup \mathbb{R}$) and a spacetime ($\mathbb{R}^3$) coordinate system. The former is a concatenation of space and time into one single frame while still keeping space and time coordinates separate. The latter, however, constitutes a coupling between space and time coordinates such that an increment in one of the coordinates yields increments both in space and time. At this point, many different local spacetime ($\mathbb{R}^3$) coordinate systems may be defined at each face in the mesh. However, since the purpose of this projection is the application of Van Leer's flux-splitting method to the spatial fluxes, the normal vector $\mathbf{n}$ to the face is taken as the first local direction

$$\mathbf{e}_1 = \mathbf{n} = \begin{Bmatrix} n_t \\ n_x \\ n_y \end{Bmatrix} \tag{21}$$

For the second and third directions there exists an infinite number of possible vectors perpendicular to $\mathbf{e}_1$ and between each other, i.e. such that $\mathbf{e}_i \cdot \mathbf{e}_j = 0$ for $i \neq j$, as required for the coordinate system to be orthogonal. Nevertheless, for the sake of simplicity, $\mathbf{e}_2$ is chosen such that it has a null component in time t. After normalisation ($||\mathbf{e}_2|| = 1$) it can be written as follows

$$\mathbf{e}_2 = \mathbf{t}_1 = \begin{Bmatrix} 0 \\ \dfrac{-n_y}{\sqrt{n_x^2 + n_y^2}} \\ \dfrac{n_x}{\sqrt{n_x^2 + n_y^2}} \end{Bmatrix} \tag{22}$$

The third direction may be defined such that the orientation of the new coordinate system is right-handed or positive, i.e.

$$\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2 \qquad \Longrightarrow \qquad \mathbf{e}_3 = \mathbf{t}_2 = \left\{ \begin{array}{c} -\sqrt{n_x^2 + n_y^2} \\ \dfrac{n_x n_t}{\sqrt{n_x^2 + n_y^2}} \\ \dfrac{n_y n_t}{\sqrt{n_x^2 + n_y^2}} \end{array} \right\} \tag{23}$$

where, again, a normalisation has been applied so that $||\mathbf{e}_3|| = 1$. As in any coordinate system transformation column vectors in matrix P are each of the unit vectors of the new base $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ written in terms of the old base's $(\mathbf{e}_t, \mathbf{e}_x, \mathbf{e}_y)$, i.e.

$$P = \left[\left\{ \mathbf{e}_1 \right\} \left\{ \mathbf{e}_2 \right\} \left\{ \mathbf{e}_3 \right\} \right] = \begin{bmatrix} n_t & 0 & -\sqrt{n_x^2 + n_y^2} \\ n_x & \dfrac{-n_y}{\sqrt{n_x^2 + n_y^2}} & \dfrac{n_x n_t}{\sqrt{n_x^2 + n_y^2}} \\ n_y & \dfrac{n_x}{\sqrt{n_x^2 + n_y^2}} & \dfrac{n_y n_t}{\sqrt{n_x^2 + n_y^2}} \end{bmatrix} \tag{24}$$

The inverse transformation matrix yields

$$P^{-1} = P^T = \begin{bmatrix} n_t & n_x & n_y \\ 0 & \dfrac{-n_y}{\sqrt{n_x^2 + n_y^2}} & \dfrac{n_x}{\sqrt{n_x^2 + n_y^2}} \\ -\sqrt{n_x^2 + n_y^2} & \dfrac{n_x n_t}{\sqrt{n_x^2 + n_y^2}} & \dfrac{n_y n_t}{\sqrt{n_x^2 + n_y^2}} \end{bmatrix} \tag{25}$$

Space fluxes calculated through equations (16)-(20) need to be projected back onto the global space coordinates, x and y. Second and third rows in matrix P, equation (24), are used for this purpose in equation (26)

$$\mathbf{f}_x \cdot n_x + \mathbf{f}_y \cdot n_y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_x & \dfrac{-n_y}{\sqrt{n_x^2 + n_y^2}} & \dfrac{n_x n_t}{\sqrt{n_x^2 + n_y^2}} & 0 \\ 0 & n_y & \dfrac{n_x}{\sqrt{n_x^2 + n_y^2}} & \dfrac{n_y n_t}{\sqrt{n_x^2 + n_y^2}} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \left\{ \begin{array}{c} \rho u_n \\ \rho u_n^2 + p \\ \rho u_n u_{t_1} \\ \rho u_n u_{t_2} \\ (\rho E + p)\, u_n \end{array} \right\} \tag{26}$$

which can be shown to yield

$$\left\{ \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho E + p)\, u \end{array} \right\} n_x + \left\{ \begin{array}{c} \rho v \\ \rho v u \\ \rho v^2 + p \\ (\rho E + p)\, v \end{array} \right\} n_y \tag{27}$$

B. Limiters - extrapolation to face values

In a central-difference scheme the value of the primitive variables at each face is worked out as an average of the values at the neighbouring cells. In an upwind scheme the direction of propagation of some quantities determine whether the values used at the cell interface are taken from the upstream or downstream neighbour cell. First-order methods use the cell-centred value of the neighbour cell as the value at the face. However, a second-order correction term may be added to the cell-centred values when extra accuracy is required. In order to avoid spurious oscillations resulting from these second-order correction terms, the so-called slope limiters φ_t, φ_x and φ_y are introduced in the current two-dimensional spacetime formulation (the superscripts f and c denote the face and cell centres, respectively)

$$\mathbf{W}^{\mathrm{f}} = \mathbf{W}^{\mathrm{c}} + \underbrace{\nabla \mathbf{W}^{\mathrm{c}} \cdot \mathbf{\Phi} \cdot \left(\mathbf{q}^{\mathrm{f}} - \mathbf{q}^{\mathrm{c}}\right)}_{\text{second-order correction term}} \tag{28}$$

where $\mathbf{q}$ is the column vector of spacetime coordinates

$$\mathbf{q} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} = \begin{Bmatrix} t \\ x \\ y \end{Bmatrix} \tag{29}$$

the matrix of slope limiters $\mathbf{\Phi}$ is defined as

$$\mathbf{\Phi} = \begin{bmatrix} \varphi_t & 0 & 0 \\ 0 & \varphi_x & 0 \\ 0 & 0 & \varphi_y \end{bmatrix} \tag{30}$$

and the matrix of gradients of the primitive variables $\nabla \mathbf{W}^{\mathrm{c}} = \dfrac{\partial \mathbf{W}^{\mathrm{c}}}{\partial \mathbf{q}}$ is given by

$$\nabla \mathbf{W}^{\mathrm{c}} = \left[\quad \frac{\partial W_i^{\mathrm{c}}}{\partial q_j} \quad \right] = \left[\left. \frac{\partial \mathbf{W}^{\mathrm{c}}}{\partial t} \right| \left. \frac{\partial \mathbf{W}^{\mathrm{c}}}{\partial x} \right| \left. \frac{\partial \mathbf{W}^{\mathrm{c}}}{\partial y} \right] = \left[\begin{array}{c|c|c} \dfrac{\partial \rho}{\partial t} & \dfrac{\partial \rho}{\partial x} & \dfrac{\partial \rho}{\partial y} \\ \dfrac{\partial u}{\partial t} & \dfrac{\partial u}{\partial x} & \dfrac{\partial u}{\partial y} \\ \dfrac{\partial v}{\partial t} & \dfrac{\partial v}{\partial x} & \dfrac{\partial v}{\partial y} \\ \dfrac{\partial p}{\partial t} & \dfrac{\partial p}{\partial x} & \dfrac{\partial p}{\partial y} \end{array} \right] \tag{31}$$

The limiters have been chosen to comply with TVD (Total Variation Diminishing) conditions [64] and depend upon the changes of the fluid variables in the nearby of the cell. To account for these changes one can define the monitor at cell j as the ratio

$$r^j = \min \left\{ \frac{\nabla W^1}{\nabla W^j}, \ldots, \frac{\nabla W^{n_{\mathrm{ne}}}}{\nabla W^j} \right\} \tag{32}$$

where n_{ne} is the number of neighbouring cells. In the specific literature there are many available methods for the computation of the limiters. One of the most well-known slope limiters is due to Van Leer [63], equation (33), and is the one used in this paper.

$$\varphi^{\mathrm{VL}}(r) = \frac{r + |r|}{1 + |r|} \tag{33}$$

V. Central-difference in Space, Upwind in Time

It was hypothesised that a step further in the development of a spacetime framework would be taking advantage of a central-difference approach in space whilst still upwinding in time. The idea underpinning this new hybrid (CSUT, namely *C*entral-difference in *S*pace, *U*pwind in *T*ime) formulation would allow the strength and robustness of the JST scheme to be retained and, at the same time, achieve more time accurate solutions, comparable to those obtained through the upwind formulation, as a consequence of the use of an appropriate time stencil. A prototype code for this formulation has been written and a small number of test cases considered will be presented. Moreover it will be shown using convergence residual plots for a number of initial test cases (for instance see Figure 8 for the simple flap deployment) that the convergence of unsteady problems is improved with respect to that of a purely JST spacetime formulation.

As done in the upwind case, the spacetime flux at any inclined face in the mesh may be split into space and time contributions. Therefore, equation (11) is still applicable in the current case where time fluxes can be worked out using equations (14)-(15). For the space fluxes, however, the usual central-difference formulation is used in this case, i.e.

$$\mathbf{f}_x\left(\mathbf{W}^{\mathrm{FACE}}\right) = \mathbf{f}_x\left(\frac{\mathbf{W}^{\mathrm{UPSTREAM}} + \mathbf{W}^{\mathrm{DOWNSTREAM}}}{2}\right) \tag{34}$$

VI. Results and Discussion

A. Periodic semi-infinite piston

As a starting point in the validation of the upwind formulation of the spacetime method a periodic semi-infinite piston is tested. This is a simple one-dimensional test case and the fact that there exists an analytical solution, equation (37), makes it a very suitable correlation. A periodicity condition is applied in time, i.e. left and right vertical boundaries on the spacetime mesh in Figure 2 are connected. The top boundary is modelled as a moving solid wall and for the bottom one, non-reflecting boundary conditions are used. The motion is sinusoidal

$$x\left(t\right) - x_0 = \frac{\Delta L}{2} \cos\left(\frac{2\pi}{T} t\right) \tag{35}$$

and the reduced frequency

$$k = \frac{\pi \Delta L}{T a_\infty} \tag{36}$$

is equal to 0.016. In equations (35)-(36), ΔL and T are the amplitude and the period of the piston's motion, respectively, and a_∞ is the speed of sound at initial conditions. This setup allows the results to be compared with piston theory at the moving wall [65]

$$\frac{p}{p_\infty} = \left(1 + \frac{\gamma - 1}{2} \frac{u_w}{a_\infty}\right)^{\frac{2\gamma}{\gamma - 1}} \tag{37}$$

Pressure contours for both the central-difference and upwind schemes are depicted in Figure 2. Also, the pressure at the moving wall is compared against theoretical results over one whole oscillation in Figure 3. These non-dimensional results correspond to a motion of amplitude $\Delta L = 10.41$ cm at 1000 rpm with sea-level ISA atmosphere conditions, i.e. $\rho_\infty = 1.225\ \text{kg} \cdot \text{m}^{-3}$ and $p_\infty = 101325$ Pa. The value of the heat capacity ratio used is $\gamma = 1.403$ and the maximum piston velocity at each cycle is $V_{\text{max}} = 5.45\ \text{m} \cdot \text{s}^{-1}$. Results for both central-difference and upwind are in good agreement with piston theory. No noticeable differences appear between the central-difference and upwind stencils which can be explained by the periodicity of the problem.

Although intuitively information can only travel forwards in time, in periodic problems information may seem to go also backwards in time thus justifying the use of a central-difference time stencil. Bearing in mind that this is just an illusion, the explanation relies on the fact that, at each cycle, the problem is influenced by any previous temporal state, hence later stages of the previous cycle (ahead in physical time domain) determine the solution at the earliest stages of the current cycle (behind in physical time domain).

B. Piston with sharp change of direction

To validate the upwind formulation of the spacetime method in a non-periodic case, in which the solution at any time level can only be influenced by the solution at previous time levels, the one-dimensional piston given by the spacetime mesh in Figure 4 is computed. Initially, up to $t = 0.4$, the piston travels downwards at a constant speed, compressing the gas inside the chamber. In contrast with problem defined in section A, the bottom boundary has been set to solid wall, leading to some wave reflections. At time $t = 0.4$ the piston suddenly inverts its velocity and from this point onwards it moves upwards at a constant speed, expanding the gas inside the chamber. The aim of this configuration is to analyse whether the upwind stencil used solves the issue of pressure waves propagating backwards in time. As shown in Figure 5, the upwind formulation improves considerably the prediction of sudden and fast movements when compared to a central-difference formulation. The typical oscillatory behaviour around shocks of JST solvers is observed here in the time direction when a sudden change in the movement of a boundary occurs. The upwind formulation damps out these oscillations successfully yielding a much smoother solution. The difference in the quality of the solutions is again explained by the fact that pressure waves always travel forwards in time. Figure 4 depicts pressure contours for this configuration.

C. Pitching aerofoil (periodic)

The first two-dimensional problem presented in this paper is a pitching NACA 0012 aerofoil simulation. The spacetime mesh is constructed from a two-dimensional structured mesh by stacking up planes in the time direction, as shown above in Figure 1. The first and last planes are connected to achieve the periodic boundary condition and an oncoming

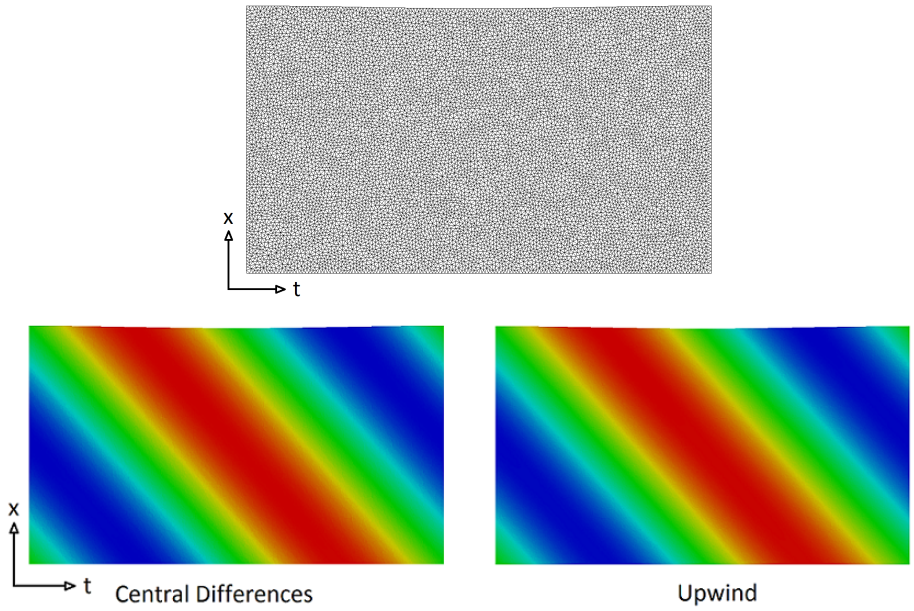


Figure 2. One dimensional periodic semi-infinite piston: spacetime mesh (top) and pressure contours (bottom)

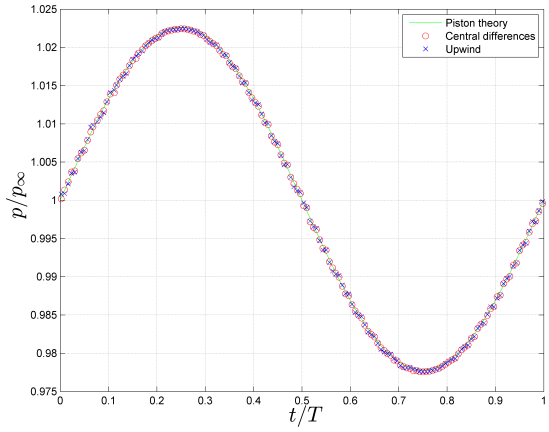


Figure 3. Comparison of central-difference and upwind results to piston theory

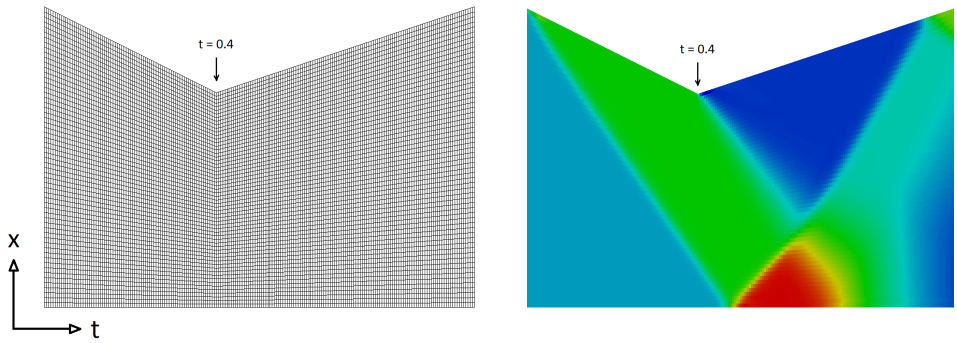


Figure 4. Spacetime mesh (left) used on one dimensional finite piston with sharp movement, and C_p contours plot (right)

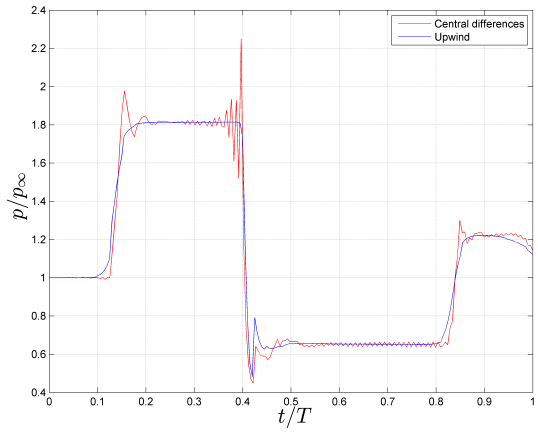


Figure 5. Comparison between central-difference and upwind results for piston with sharp movement

flow is set with a freestream velocity $M_\infty = 0.85$. The aerofoil oscillates at a reduced frequency of $k = \frac{\omega c}{2U_\infty} = 0.0814$ about its quarter chord following the sinusoidal movement

$$\alpha = \alpha_0 + \Delta\alpha \sin \omega t \tag{38}$$

where $\alpha_0 = 0$ deg and $\Delta\alpha = 2.51$ deg. C_p distribution plots at $\omega t = 0$, $\omega t = \frac{2\pi}{3}$ and $\omega t = \frac{4\pi}{3}$ have been depicted in Figure 6. The CSUT (*C*entral-difference in *S*pace, *U*pwind in *T*ime) and JST (central-difference) formulations yield very similar C_p distributions, likely down to the periodicity of the problem (note that the only difference between CSUT and JST comes from the time stencil). On the other hand, the upwind formulation gives a slightly different solution, probably due to a higher order scheme used in space. Given the periodicity of the problem all three methods converge at comparable rates, as depicted in Figure 7, although the upwind method still produces a slightly better convergence history as a consequence of its higher order nature.

D. Simple flap

Using RBF's to deform the two dimensional mesh and stacking up planes in the time direction, as done in the case of the pitching aerofoil (section C), a spacetime mesh is created to simulate the deflection of a simple flap on a NACA-0012 (Figure 8). Initially the aerofoil is flying with an angle of attack $\alpha = 0$ deg at $M_\infty = 0.7$. After some time the flap deflects an angle $\Delta\theta = 13.5$ deg at a reduced frequency $k = \frac{\omega b}{U_\infty} = 0.375$, where ω is the angular velocity, b is the semi-chord and U_∞ is the freestream velocity. An unsteady solution for the transient process is sought and all three formulations are used to solve it. C_p contour plots and distributions along the chord at various time levels/slices are depicted in Figures 9 and 10, respectively. The hybrid CSUT formulation avoids the non-physical behaviour of the JST in the transient part (time slices 1, 2 and 3) by the use of a more realistic time stencil and matches the solution of the upwind scheme better. As concluded from slice 4 in Figure 10, the CSUT steady state solution resembles the central-difference scheme more closely since time fluxes have a negligible impact on it and only space fluxes, which are worked out using central-differences in both cases, have an effect on the steady state solution. Moreover, both the upwind and CSUT solutions converge much faster than the JST due to no pressure waves moving backwards in time, as demonstrated by convergence residuals in Figure 8. This represents a significant improvement and demonstrates the applicability of the method to transient problems.

E. Spoiler

The mesh generation of two-dimensional problems like the pitching aerofoil (section C), or even the simple flap deflection (section D), in spacetime may be done using a three-dimensional structured mesh generator or simply by stacking up two-dimensional meshes. However, in the case of more complex and arbitrary boundary motions the use of unstructured meshes is crucial. Figure 11 shows the unstructured spacetime mesh used in the solution of the current spoiler deployment case. Initially a NACA 0012 aerofoil at an angle of incidence of $\alpha = 0$ deg is immersed in a flow at $M_\infty = 0.25$. At some point, a spoiler deflects up to an angle of $\theta = 45$ deg at a reduced frequency $k = \frac{\omega b}{U_\infty} = 0.262$,

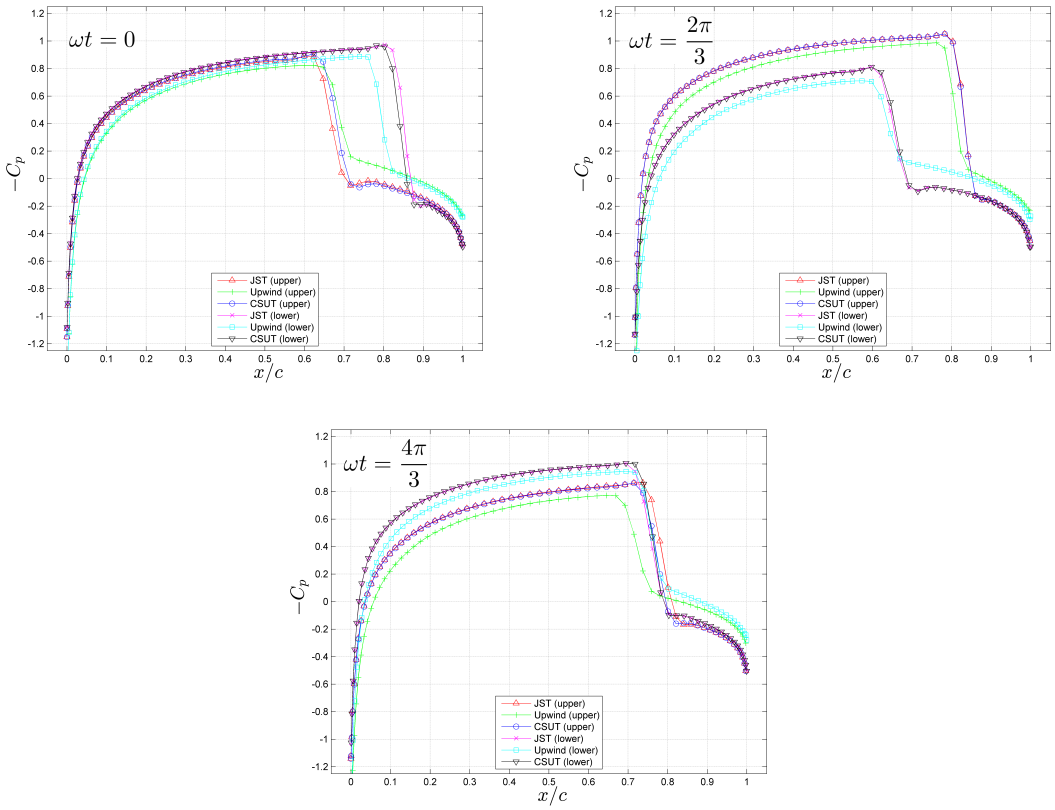


Figure 6. C_p **distributions for pitching NACA 0012 at** $M_\infty = 0.85$ **for** $\omega t = 0, \frac{2\pi}{3}, \frac{4\pi}{3}$

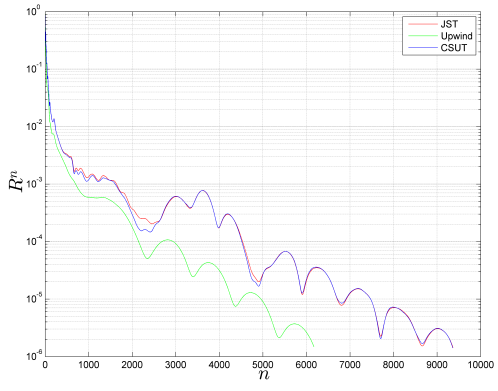


Figure 7. Convergence residuals for pitching NACA 0012 at $M_\infty = 0.85$

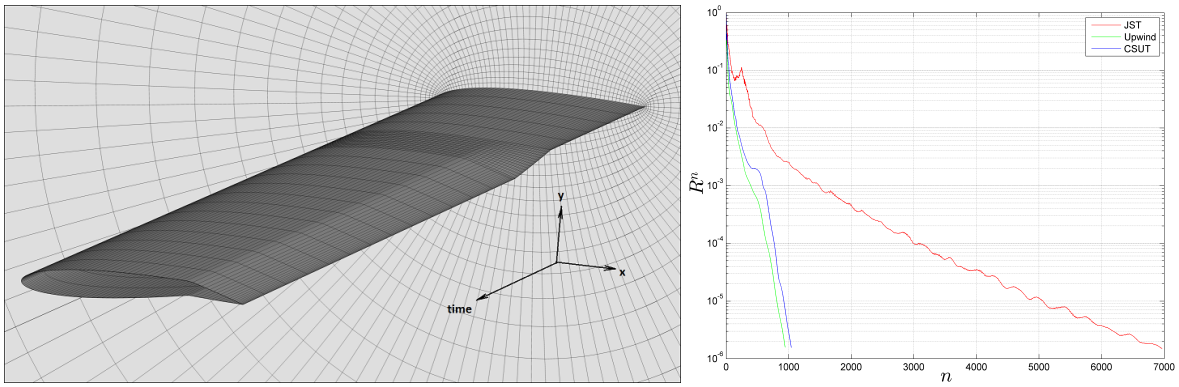


Figure 8. Simple flap deflection on a NACA 0012 aerofoil flying at $M_\infty = 0.7$: spacetime mesh (*left*) and convergence residuals (*right*)

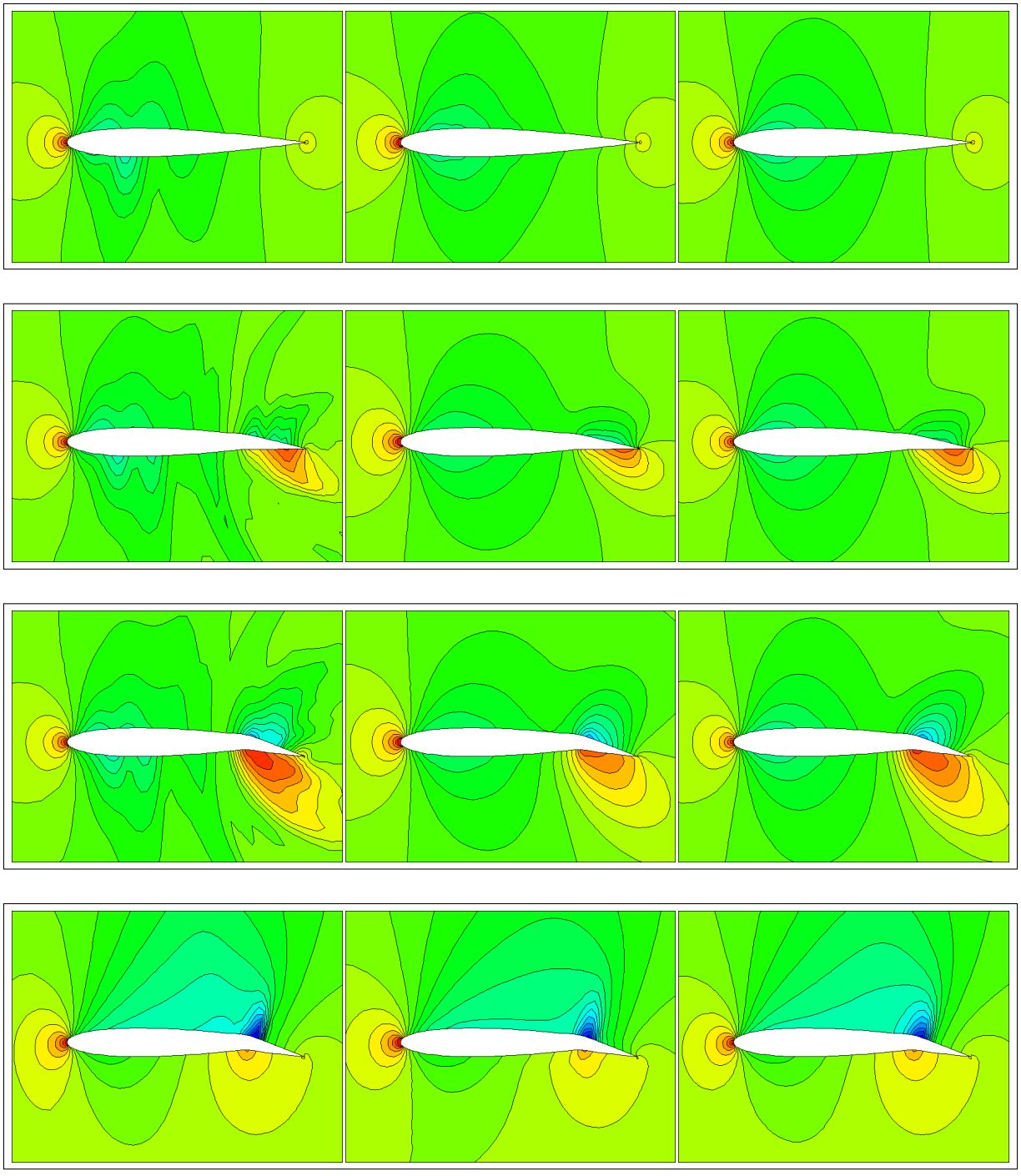
and the whole transient process is successfully captured. As in previous sections, pressure contour plots are depicted in Figure 12 for several time levels and convergence residuals are given in Figure 13. The upwind simulation converges much faster than the central-difference counterpart, as would be expected from the fact that it uses a more realistic time stencil. The convergence history of the CSUT formulation lies in between, improving the convergence of the central-difference but, at the same time, producing a more physically relevant solution, closer to that of the upwind formulation.

F. Landing

One of the main benefits of the spacetime formulation is its versatility and the fact that it is capable of handling very complex boundary motions with relative ease. Problems like a slat and flap deployment can be solved without the need for further modifications to the solver nor the implementation of intricate interpolation methods that connect cells between different time levels. With conventional time-marching techniques, defining flow properties at positions where there was not fluid at a former time level can sometimes be problematic. Moreover, this relies heavily on the accuracy of the interpolation method used. The use of a finite-volume method, conservative by nature, in space and time simplifies the problem considerably.

A simplified version of all the motions that a wing undergoes during approaching and landing has been modelled here, i.e. a slat and flap deployment on an aerofoil flying at $M_\infty = 0.15$ followed by an increase of its angle of incidence and a spoiler deployment which, in turn, decreases the incidence, all of which happens while approaching to the ground. Figure 14 shows the spacetime mesh used to represent the geometry for this problem and define the motions involved in it. As mentioned above, the solver can be left unchanged speeding up the overall simulation process, from meshing through to running the CFD code. Pressure contour plots with streamlines have been depicted in Figures 15 and 16 for all three formulations (upwind, central-difference and CSUT) and the history of convergence residuals is plotted in Figure 13. In order to understand what the streamlines represent it is important to realize that the reference frame chosen for this simulation is not fixed to the aerofoil nor the ground; it moves with the aerofoil on the horizontal direction but remains fixed in the vertical direction, i.e. null vertical velocity.

Unexpectedly, the rate of convergence of the hybrid solver is not much better (or at least the difference is negligible) than that of the central-difference solver, as can be observed in Figure 13. A possible explanation for this is that the gradients of the fluid properties are worked out, as usual, using a central-difference approach, despite the use of an upwind stencil in the time direction. Time fluxes are therefore slightly affected by future events. Although this is a minor contribution it may become more noticeable when the solution is unstable and changes rapidly.



(a) (b) (c)

Figure 9. C_p **contours for simple flap deflection at** $M_\infty = 0.7$**:** (a) **central-difference (left),** (b) **upwind (centre), and** (c) **central-difference in space, upwind in time (right)**

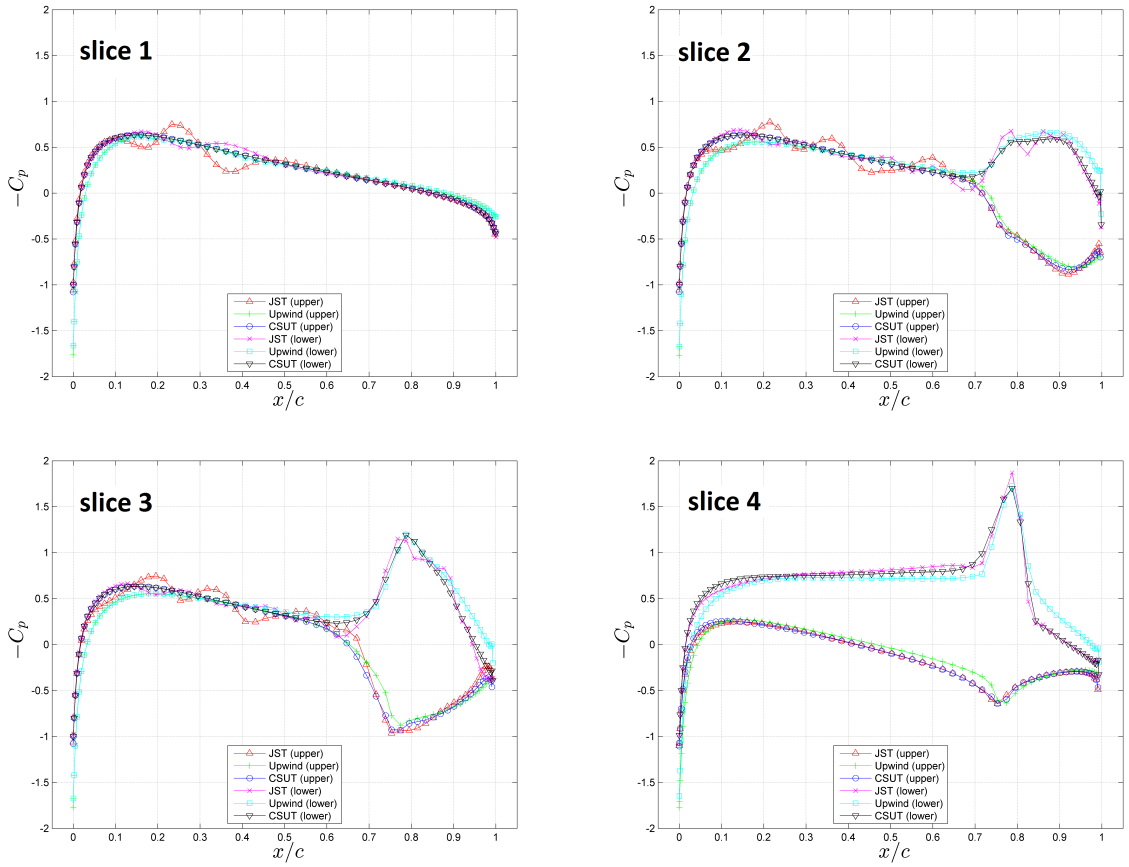


Figure 10. C_p **distributions at four different time slices corresponding to those in Figure 9 for simple flap deflection at** $M_\infty = 0.7$

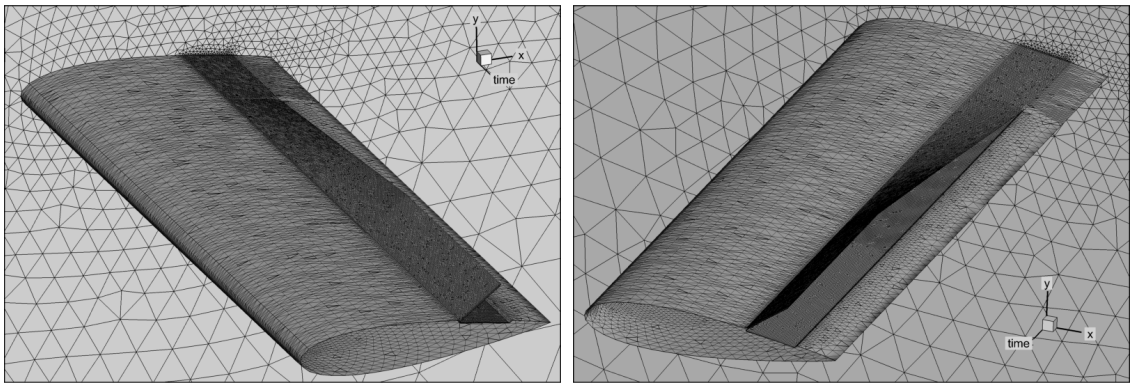


Figure 11. Spacetime mesh for spoiler deployment

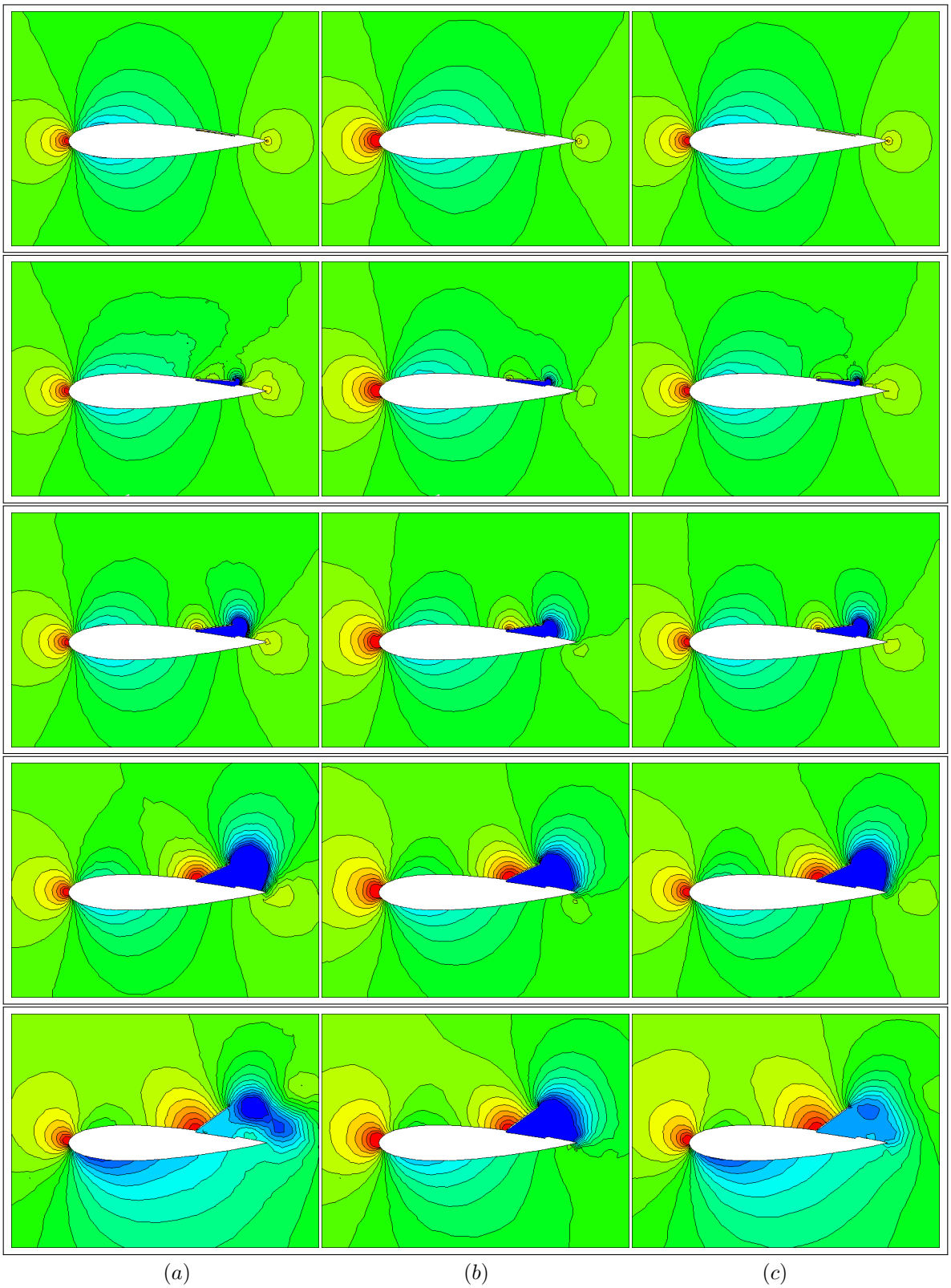


Figure 12. C_p **contours for spoiler deployment at** $M_\infty = 0.25$**:** (a) **central-difference (left),** (b) **upwind (centre) and** (c) **central-difference in space, upwind in time (right)**

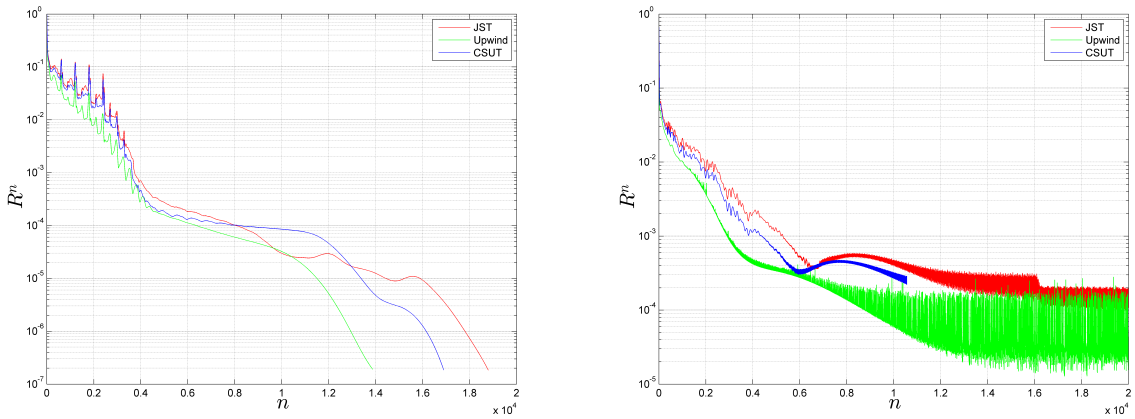


Figure 13. Convergence residuals: spoiler deployment at $M_\infty = 0.25$ (*left*) and landing case at $M_\infty = 0.15$ (*right*)

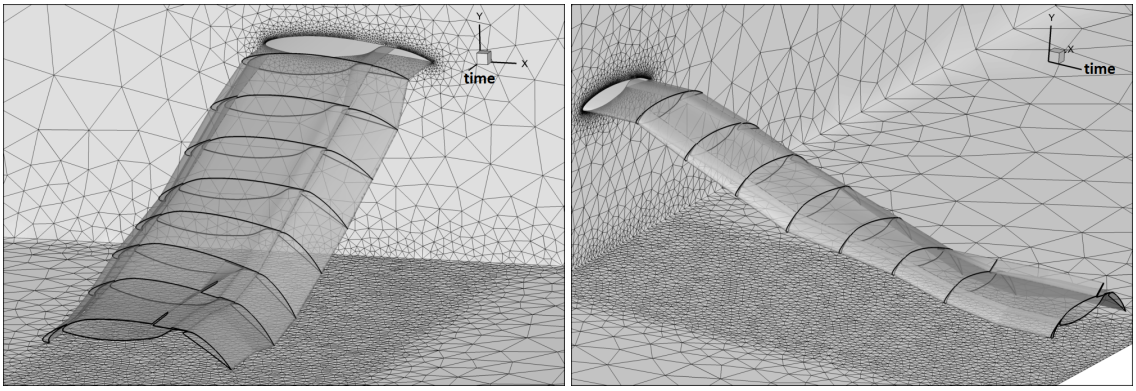


Figure 14. Spacetime mesh for landing case, i.e. aerofoil with a slat and flap deployment followed by an increase of its angle of incidence and a spoiler deployment which, in turn, decreases the incidence, all of which happens while approaching to the ground.

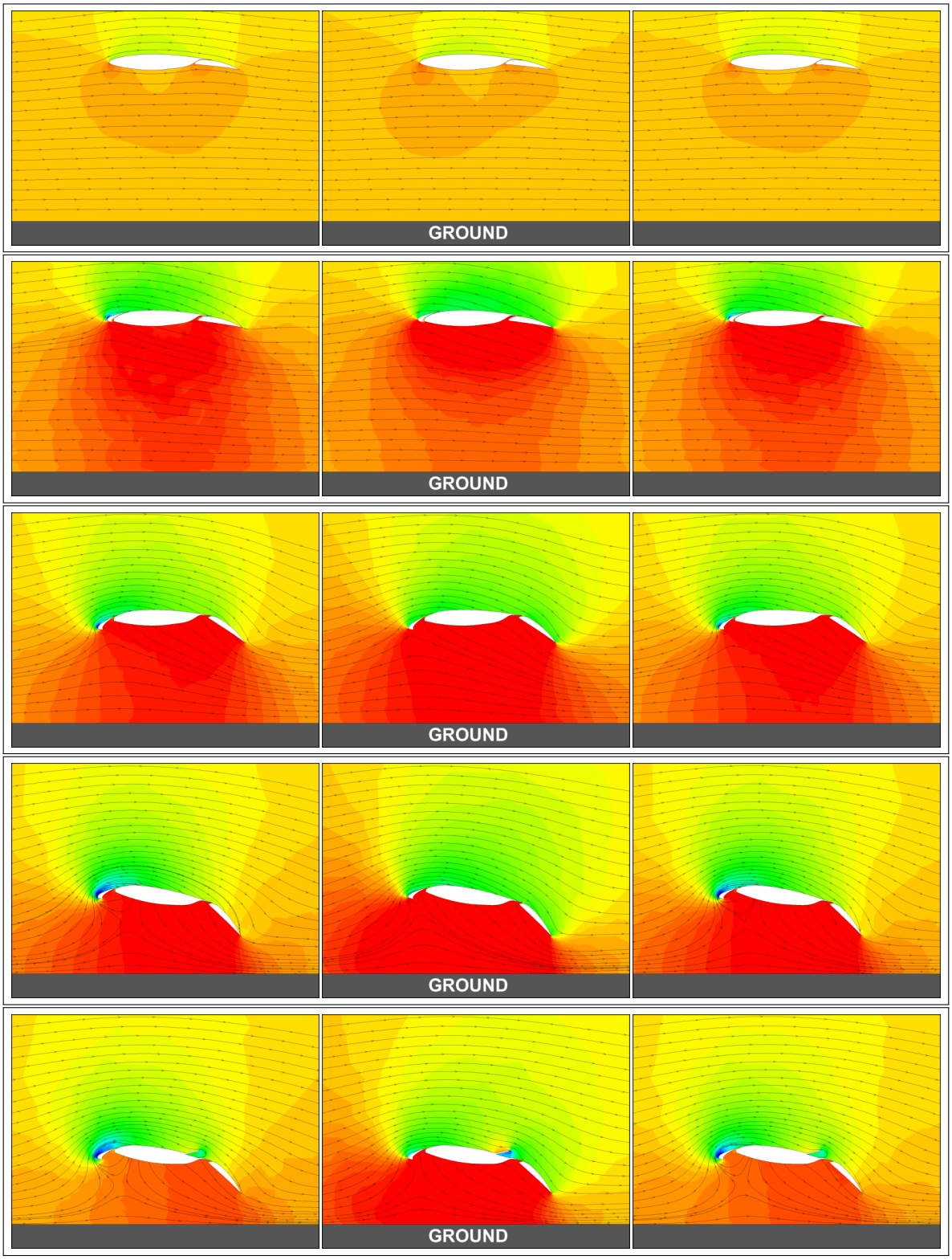


Figure 15. C_p **contours and streamlines for landing case at** $M_\infty = 0.15$**:** (a) **central-difference (left),** (b) **upwind (centre) and** (c) **central-difference in space, upwind in time (right).** ***Continues in Figure 16***

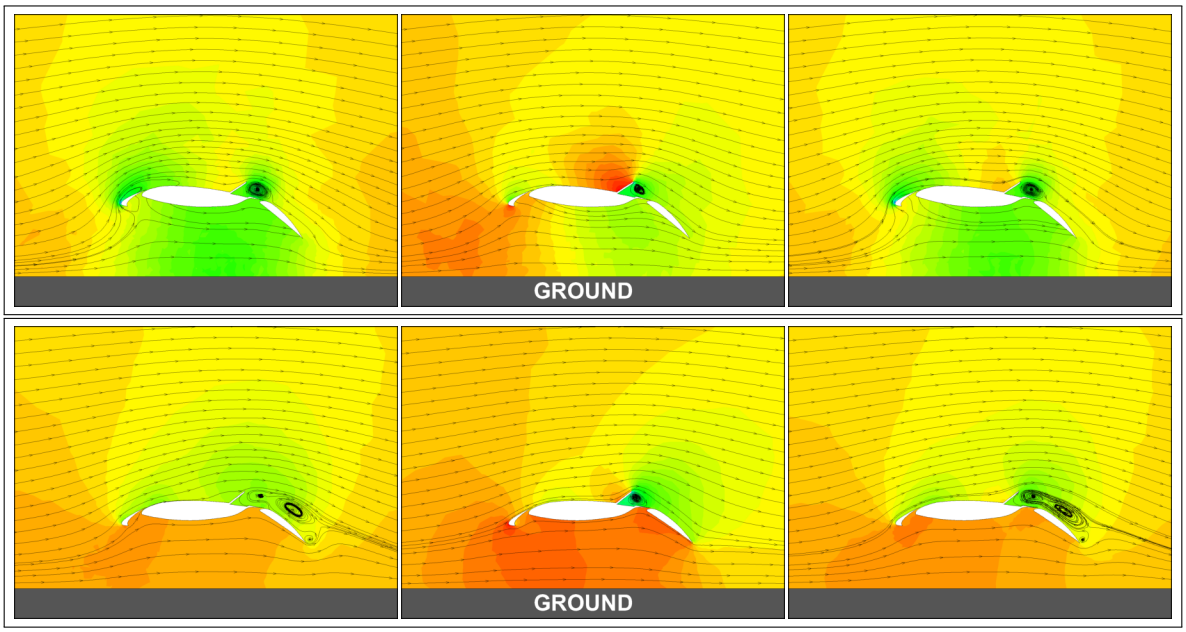


Figure 16. ***Continuation from Figure 15.*** C_p **contours and streamlines for landing case at** $M_\infty = 0.15$**:** (a) **central-difference (left),** (b) **upwind (centre) and** (c) **central-difference in space, upwind in time (right).**

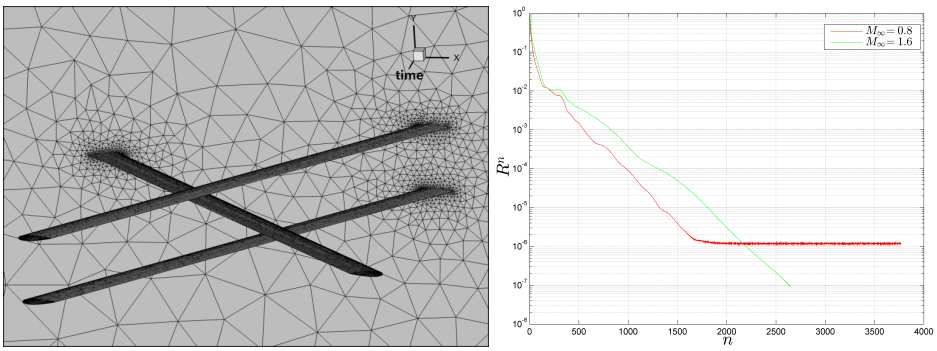


Figure 17. Flying aerofoils in opposite directions: spacetime mesh (left) and convergence residuals (right)

G. Flying aerofoils in opposite directions

The last problem serves as a final proof of the versatility that this method brings to unsteady aerodynamics simulations. The complex problem of two parallel aerofoils flying towards another aerofoil that flies between them in the opposite direction is effectively solved with ease by the spacetime method. No further modifications to the solver are required and no mesh interpolations are needed as it would otherwise be the case with a conventional dual time-stepping technique. The spacetime geometry used for this problem is depicted in Figure 17 along with the fluid domain. One of the key points of this kind of simulation is the fact that, as opposed to conventional methods, the solution is worked out in a fixed reference frame, i.e. the freestream condition is $M_\infty = 0$. In other words, instead of the fluid moving towards the aerofoils, the aerofoils themselves move at a certain speed with respect to the static air. This simplifies greatly problems like moving apart/closer objects that would have otherwise been calculated via intricate mesh adaptation and interpolation techniques such as Chimera grids. Pressure contour plots at different time levels are given in Figure 18 for the aerofoils moving at $M = 0.8$ and $M = 1.6$.

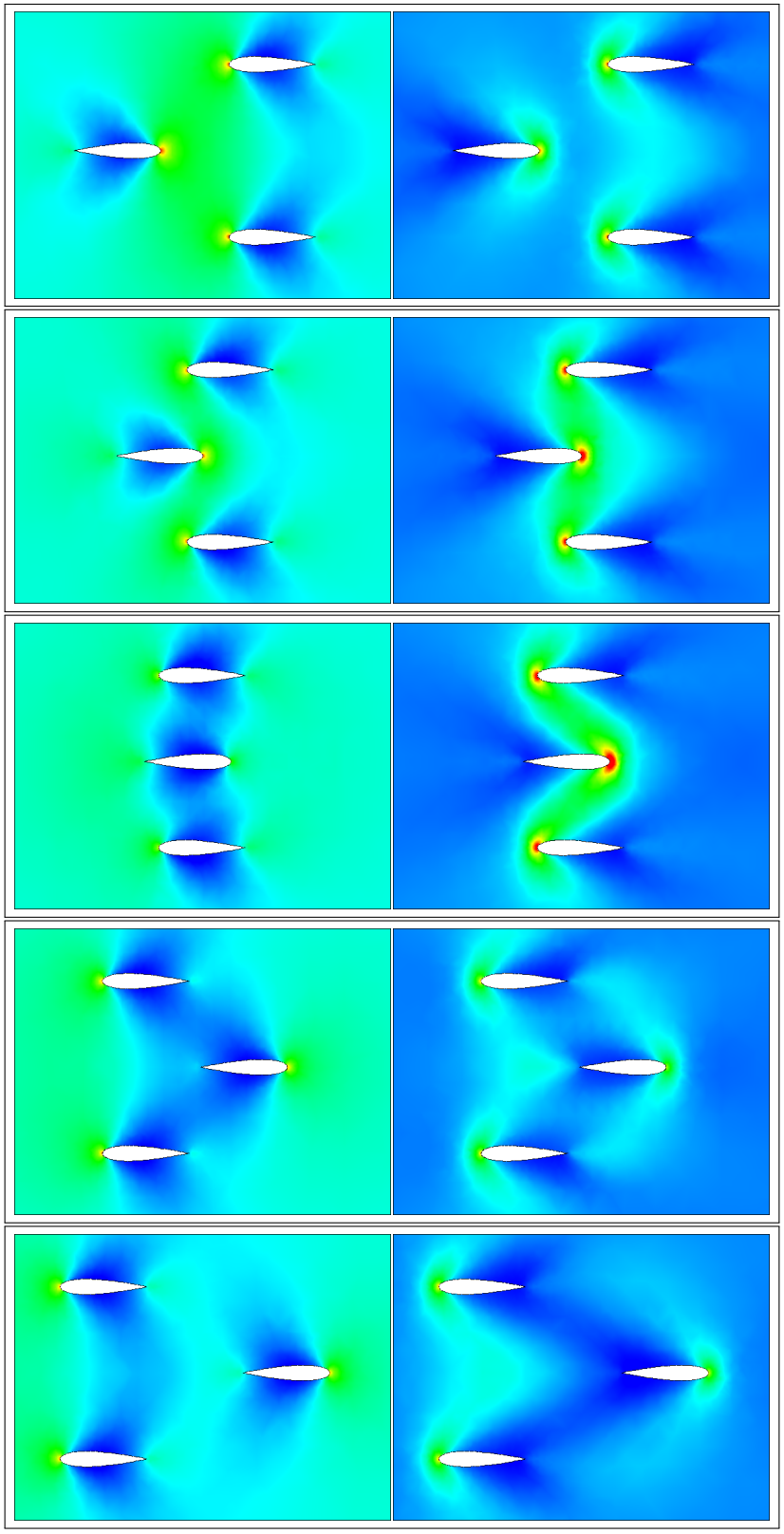
Another potential problem, and similar in complexity, would be the case of a rotor-stator interaction. This is particularly well-suited for the spacetime framework as it is periodic both in space and time.

VII. Conclusions

The applicability and versatility of the spacetime framework in unsteady aerodynamics problems has been successfully proven in the current paper. Due to its conservative nature in space and, particularly, in time, spacetime is very well-suited for periodic problems where initial and final states are connected. It is safe to state that a solution is converged once residuals have dropped below a certain threshold. Moreover, the use of a coupled finite-volume formulation in space and time allows to define different timestep sizes at different locations, being possible, for instance, the use of a bigger timestep far away from the aerofoil while still keeping a small timestep close to the aerofoil where changes in the fluid properties happen faster. This can improve the efficiency of simulations.

Industrial applications could benefit substantially from the use of the spacetime framework due to its potential for highly automated CFD simulations which would, in turn, speed up the design cycle. It has been shown that the solver can be left unchanged throughout the whole range of problems described in this work. Besides, there is no need to implement intricate interpolation methods, which can lead to losses in accuracy, in order to connect cells between different time levels. Spacetime solvers cope well with appearing/disappearing cells which brings a great flexibility to the method.

It has been demonstrated that the upwind formulation of the spacetime framework for unsteady problems yields more representative solutions than the central-difference counterpart. This is likely due to the use of a more realistic time stencil where pressure waves are not allowed to travel backwards in time. The central-difference and hybrid formulations have been found consistently more sensitive to the freestream velocity and rate of change of boundary movements than the upwind one. This is also noticeable in the rate of convergence of the simulations where the upwind solver outperforms the hybrid one, which, in general, converges faster than the central-difference counterpart.



(a) (b)

Figure 18. C_p **contours for aerofoils flying in opposite directions at:** (a) $M_\infty = 0.8$ **(left) and** (b) $M_\infty = 1.6$ **(right)**

References

[1] Thomas C. S. Rendall, Christian B. Allen, and Edward D. C. Power. Conservative unsteady aerodynamic simulation of arbitrary boundary motion using structured and unstructured meshes in time. *International Journal for Numerical Methods in Fluids*, 70(12):1518–1542, dec 2012.

[2] C.W Hirt, A.A Amsden, and J.L Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, mar 1974.

[3] Giridhar Jothiprasad, Dimitri J. Mavriplis, and David A. Caughey. Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes. *Journal of Computational Physics*, 191(2):542–566, nov 2003.

[4] Zhi Yang and Dimitri J. Mavriplis. Higher-Order Time Integration Schemes for Aeroelastic Applications on Unstructured Meshes. *AIAA Journal*, 45(1):138–150, jan 2007.

[5] Dimitri J Mavriplis. *Mesh generation and adaptivity for complex geometries and flows*. Elsevier, 1996.

[6] John T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, 28(8):1381–1388, aug 1990.

[7] Cizmas P. and J I Gargoloff. Mesh Generation and Deformation Algorithm for Aeroelastic Simulations. *Journal of Aircraft*, 45(3):1062–1066, may 2008.

[8] R. Duvigneau and M Visonneau. Shape optimization of incompressible and turbulent flows using the simplex method. In *15th AIAA Computational Fluid Dynamics Conference*, number AIAA 2001-2533, Reston, Virigina, jun 2001. American Institute of Aeronautics & Astronautics, American Institute of Aeronautics and Astronautics.

[9] C. Farhat, C. Degand, B. Koobus, and M Lesoinne. An improved method of spring analogy for dynamic unstructured fluid meshes. *39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, (AIAA 1998-2070), apr 1998.

[10] William J. Gordon and Charles A Hall. Construction of curvilinear co-ordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, 7(4):461–477, 1973.

[11] William J. Gordon and Linda C Thiel. *Transfinite mappings and their application to grid generation*, volume 10-11. Elsevier, jan 1982.

[12] Martin D Buhmann and Martin D Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[13] Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, 2004.

[14] Christian Allen and Thomas Rendall. Unified Approach to CFD-CSD Interpolation and Mesh Motion Using Radial Basis Functions. *25th AIAA Applied Aerodynamics Conference*, (AIAA 2007-3804), jun 2007.

[15] T. C. S. Rendall and C B Allen. Unified fluidstructure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559, jun 2008.

[16] Leevan Ling and Robert Schaback. Stable and Convergent Unsymmetric Meshless Collocation Methods. *SIAM Journal on Numerical Analysis*, 46(3):1097–1115, jan 2008.

[17] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. Geometric numerical integration for complex dynamics of tethered spacecraft. *Proceedings of the 2011 American Control Conference*, (March):1885–1891, 2011.

[18] Scott A Sarra and Edward J Kansa. Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations, 2009.

[19] D A Field. Laplacian Smoothing And Delaunay Triangulations. *Communications in Applied Numerical Methods*, 4:709–712, 1988.

[20] L A Freitag. On Combining Laplacian and Optimization-based Smoothing Techniques. *Trends in Unstructured Mesh Generation*, 220:37–44, 1997.

[21] Dougherty F. C., Benek J. A., Joseph L Steger, F Carroll. Dougherty, John A Benek, Joseph L Steger, and Ames Research Center. *On applications of Chimera grid schemes to store separation*. National Aeronautics and Space Administration, Ames Research Center Moffett Field, Calif, oct 1985.

[22] Robert Meakin and Robert Meakin. On adaptive refinement and overset structured grids. *13th Computational Fluid Dynamics Conference*, (AIAA 1997-1858):236–249, jun 1997.

[23] Piergiovanni Renzoni, Alessandro D'Alascio, Norbert Kroll, Dave Peshkin, Michael H.L. Hounjet, Jean-Christophe Boniface, Luigi Vigevano, Christian B. Allen, Ken Badcock, Lorenzo Mottura, Eberhard Schöll, and Anastasios Kokkalis. EROS a common European Euler code for the analysis of the helicopter rotor flowfield. *Progress in Aerospace Sciences*, 36(5-6):437–485, aug 2000.

[24] Hubert Pomin and Siegfried Wagner. Aeroelastic Analysis of Helicopter Rotor Blades on Deformable Chimera Grids. *Journal of Aircraft*, 41(3):577–584, may 2004.

[25] Christopher L Rumsey. Computation of Acoustic Waves Through Sliding-Zone Interfaces. *AIAA Journal*, 35(2):263–268, feb 1997.

[26] R. Steijl and G Barakos. Sliding mesh algorithm for CFD analysis of helicopter rotor-fuselage aerodynamics. *International Journal for Numerical Methods in Fluids*, 58(5):527–549, oct 2008.

[27] C L Fenwick and C B Allen. Development and Validation of Sliding and Non-Matching Grid Technology for Control Surface Representation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 220(4):299–315, jan 2006.

[28] Lai M. C. and C S Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *Journal of Computational Physics*, 160:705–719, 2000.

[29] C S Peskin. The immersed boundary method. *Acta Numerica*, pages 479–517, 2002.

[30] Ya'eer Kidron, Yair Mor-Yossef, and Yuval Levy. Robust Cartesian Grid Flow Solver for High-Reynolds-Number Turbulent Flow Simulations. *AIAA Journal*, 48(6):1130–1140, jun 2010.

[31] Matthias Meinke, Lennart Schneiders, Claudia Günther, and Wolfgang Schröder. A cut-cell method for sharp moving boundaries in Cartesian grids. *Computers & Fluids*, 85:135–142, oct 2013.

[32] M. D. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, and M. Napolitano. An immersed boundary method for compressible flows using local grid refinement. *Journal of Computational Physics*, 225(2):2098–2117, 2007.

[33] A J Katz. *Meshless Methods For Computational Fluid Dynamics*. Phd thesis, Stanford University - Department of Aeronautics and Astronautics, jan 2009.

[34] Ortega E., Onate E., Idelsohn S., and R Flores. A meshless finite point method for three-dimensional analysis of compressible flow problems involving moving boundaries and adaptivity. *International Journal for Numerical Methods in Fluids*, 73:323–343, 2013.

[35] J.J Monaghan and R.A Gingold. Shock simulation by the particle method SPH. *Journal of Computational Physics*, 52(2):374–389, nov 1983.

[36] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, and R. L. Taylor. A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow. *International Journal for Numerical Methods in Engineering*, 39(22):3839–3866, nov 1996.

[37] Netuzhylov H. and A Zilian. Space-time meshfree collocation method: methodology and application to initial-boundary value problems. *International Journal for Numerical Methods in Engineering*, 80:355–380, 2009.

[38] Formaggia L., Peraire J., and K Morgan. Simulation of a store separation using the finite element method. *Applied Mathematical Modelling*, 12:175–181, apr 1988.

[39] Scott M. Murman, Michael J. Aftosmis, and Marsha J Berger. Simulations of Store Separation from an F/A-18 with a Cartesian Method. *Journal of Aircraft*, 41(4):870–878, jul 2004.

[40] Probert E. J., Hassan O., Morgan K., and J Peraire. An Adaptive Finite Element Method For Transient Compressible Flows With Moving Boundaries. *International Journal for Numerical Methods in Engineering*, 32:751–765, 1991.

[41] MICHAEL GILES. Stator/rotor interaction in a transonic turbine. *24th Joint Propulsion Conference*, (AIAA 1988-3093), jul 1988.

[42] Hughes T. J. R. and G M Hulbert. Space-Time Finite Element Methods For Elastodynamics: Formulations And Error Estimates. *Computer Methods in Applied Mechanics and Engineering*, 66:339–363, 1988.

[43] Robert B. Lowrie, Philip L. Roe, and Bram van Leer. Space-Time Methods for Hyperbolic Conservation Laws. Proceedings of ICASE/LaRC LA-UR-96-3705, Los Alamos National Laboratory, Langley Research Center, Hampton, Virginia, aug 1998.

[44] Lonny Lonny Thompson and Dantong He. Local Space-Time Adaptive Discontinuous Galerkin Finite Element Methods for Time-Dependent Waves. *Noise Control and Acoustics*, 2003(IMECE2003-42542):81–95, nov 2003.

[45] S. Ray. A model of the interaction of a fluid with multiple deformable bodies. *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, (AIAA Paper 1998-3155), jul 1998.

[46] Hsin-Hua Tsuei, Biing-Horng Liou, and S. Yu. Direct calculation of turbomachinery flows using the space-time conservation element and solution element method. *39th Aerospace Sciences Meeting and Exhibit*, (AIAA 2001-0531), jan 2001.

[47] Sin-Chung Chang. New developments in the method of space-time conservation element and solution element: Applications to the Euler and Navier-Stokes equations. Technical Memorandum NASA-TM-106226, NASA Lewis Research Center, Cleveland, OH, United States, jul 1993.

[48] Chang S. C., Wang X. Y., and C Y Chow. New Developments in the Method of Space-Time Conservation Element and Solution Element - Applications to Two-Dimensional Time-Marching Problems. Technical Memorandum NASA-TM-I06758, NASA, dec 1994.

[49] Sin-Chung Chang. The Method of Space-Time Conservation Element and Solution ElementA New Approach for Solving the Navier-Stokes and Euler Equations. *Journal of Computational Physics*, 119(2):295–324, jul 1995.

[50] Ray Hixon. Space-Time Mapping Analysis for the Accurate Calculation of Complex Unsteady Flows. *9th AIAA/CEAS Aeroacoustics Conference and Exhibit*, (AIAA 2003-3205), may 2003.

[51] R Hixon. Method and system for the efficient calculation of unsteady processes on arbitrary space-time domains.

[52] Golubev V. V., Mankbadi R. R., and R Hixon. Space-Time Mapping Analysis of Airfoil Nonlinear Interaction with Unsteady Inviscid Flow. *AIAA Journal*, 43(10):2147–2156, oct 2005.

[53] Vladimir Golubev and Axel Rohde. Application of Space-Time Mapping Analysis Method to Unsteady Nonlinear Gust-Airfoil Interaction Problem. *16th AIAA Computational Fluid Dynamics Conference*, (AIAA 2003-3693), jun 2003.

[54] Zwart P. J., Raithby G. D., and M J Raw. The Integrated Space-Time Finite Volume Method and Its Application to Moving Boundary Problems. *Journal of Computational Physics*, 154:497–519, 1999.

[55] H. van der Ven. An adaptive multitime multigrid algorithm for time-periodic flow simulations. *Journal of Computational Physics*, 227(10):5286–5303, may 2008.

[56] Edward Power, Thomas Rendall, and Christian Allen. Evaluation of a Spacetime Formulation of the Euler Equations. *30th AIAA Applied Aerodynamics Conference*, (AIAA 2012-3032), jun 2012.

[57] Thomas Rendall and Christian Allen. Conservative Unsteady Simulation of Arbitrary Boundary Deformation Using Spacetime Meshes. *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, (AIAA 2010-509), jan 2010.

[58] Thomas Rendall and Christian Allen. CFD Simulation of Arbitrary Motion in Two-Dimensional Spacetime Using Cut-Cell Meshes. *28th AIAA Applied Aerodynamics Conference*, (AIAA 2010-4696), jun 2010.

[59] M Behr. Simplex Space-Time Meshes in Finite Element Simulations. *International Journal for Numerical Methods in Fluids*, 57:1421–1434, 2008.

[60] Ungor A. and A Sheffer. Tent-Pitcher: A Meshing Algorithm For Space-Time Discontinuous Galerkin Methods. *Proceedings 9th International Meshing Roundtable*, pages 111–122, 2000.

[61] Abedi R., Chung S. H., Erickson J., Fan Y., Garland M., Guoy D., Haber R., Sullivan J. M., Thite S., and Y Zhou. Spacetime Meshing with Adaptive Refinement and Coarsening. *Proceedings of the 20th Annual ACM Symposium on Computational Geometry*, pages 300–309, 2004.

[62] A. L. Gaitonde. A dual-time method for the solution of the unsteady Euler equations. *The Aeronautical Journal*, 98(978):283–291, oct 1994.

[63] B Van Leer. Flux-Vector Splitting for the Euler Equations. In *8th International Conference on Numerical Methods in Fluid Dynamics*, jun 1982.

[64] Sung-Ik Sohn. A new TVD-MUSCL scheme for hyperbolic conservation laws. *Computers & Mathematics with Applications*, 50(1-2):231–248, jul 2005.

[65] Ascher H. Shapiro. *The Dynamics And Thermodynamics Of Compressible Fluid Flow*. 1954.